GEBZE TEKNİK ÜNİVERSİTESİ

ELECTRONICAL ENGINEERING

MAT214 NUMERICAL METHODS

2020 – 2021 FALL

PROJECT 2

Student's

Number :1901022255

Name - Surname :Emirhan Köse

ABSTRACT

In this Project there were 4 current-time text files. Time and currents values have read from the files and used in the Project fort he find of RL circuits voltage. First current1.dat has read and time,current values applied the forward difference, backward difference and three point formula. After, same steps has applied fort he other time -current datas and graphics have made. Converges analysis have made about the all of datas and some comments have made about the results.

INTRODUCTION

In this Project the problem is find the RL circuits voltage approximately. While the voltages finding forward difference method, backward difference method and threepoint formula have applied separately. Fort he converges of the voltages norm method has used. While making plots loglog and plot methods have used.

READING THE FILES.

```
clc;
clear;
format long;
file=fopen('current1.dat','r');   %files have opened..
file2=fopen('current2.dat','r');
file3=fopen('current3.dat','r');
file4=fopen('current4.dat','r');
current1=fscanf(file,'%f %f',[2 inf]);    %files have read...
current2=fscanf(file2,'%f %f',[2 inf]);
current3=fscanf(file3,'%f %f',[2 inf]);
current4=fscanf(file4,'%f %f',[2 inf]);
```

FORWARD DIFFERENCE

Forward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Formula used in forward difference method.

How to used in code.

```
%current1 forward-backward-midpoint
h=current1(1,2)-current1(1,1);  % change of time of current1.dat
fdrv1=[];
for i=2:size(current1,2)             % size(current1,2)=9   so i=1 from to 9
  der=(current1(2,i)-current1(2,i-1))/h;
  fdrv1(i)=der;
end
```

BACKWARD DIFFERENCE

Backward Difference Formula (1st order accurate)

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

Formula used in backward difference

How to used in code.

```
h=current1(1,2)-current1(1,1);
bdrv1=[];
for i=2:size(current1,2)
  der=(current1(2,i)-current1(2,i-1))/h;
  bdrv1(i)=der;
end
```

FIRST POINT FORMULA-END POINT FORMULA

$$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_1) - f(x_2)]$$

$$f'(x_2) = \frac{1}{2h}[f(x_0) - 4f(x_1) + 3f(x_2)]$$

Formula used in first point formula.

How to used in code.

```
mdrv1(1)=(-(3*current1(2,1))+(4*current1(2,2))-current1(2,3))/(2*h);
mdrv1(size(current1,2))=(current1(2,1)-4*current1(2,2)+(3*current1(2,3)))/(2*h);
```

MIDPOINT FORMULA

$$f'(x_1) = \frac{1}{2h}[-f(x_0) + f(x_2)]$$

Formula used in midpoint .

How to used in code.

```
for i=2:size(current1,2)-1
    der=(current1(2,i+1)-current1(2,i-1))/(2*h);
    mdrv1(i)=der;
end
```

FINDING VOLTAGE

$$\mathcal{E}(t) = L\frac{d}{dt}i(t) + Ri(t),$$

Formula used in finding voltage

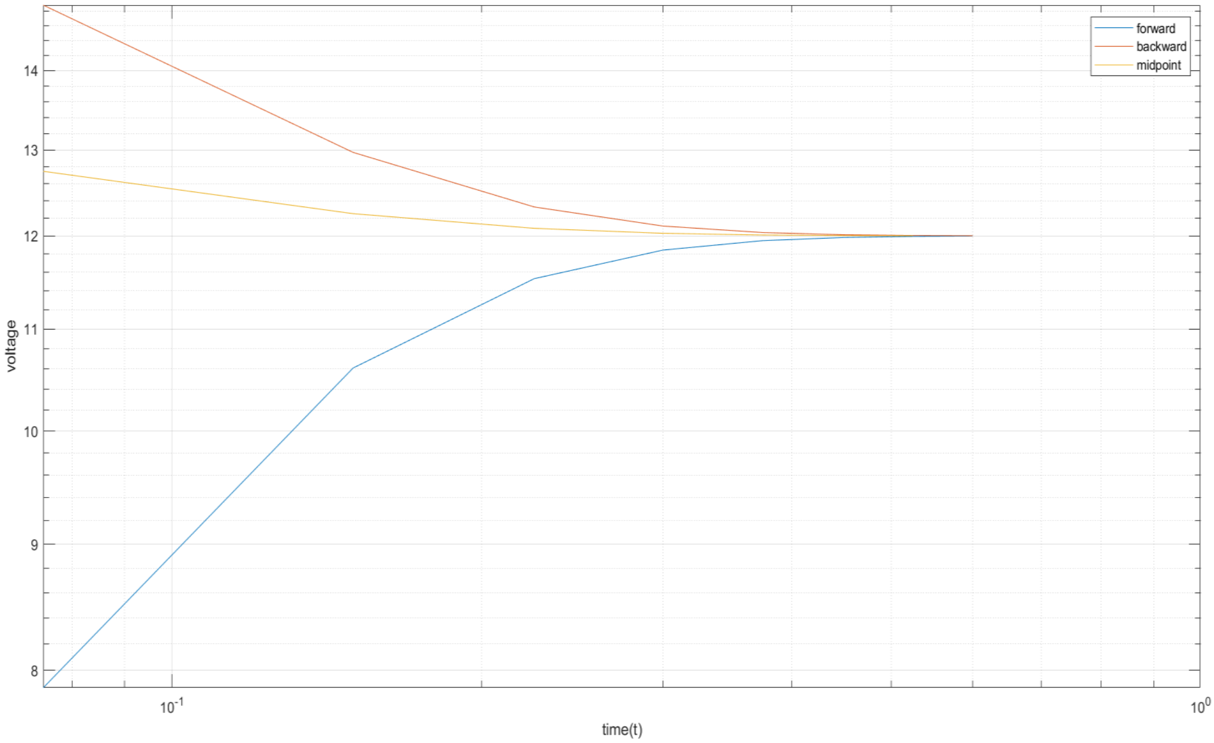
How to used in code.

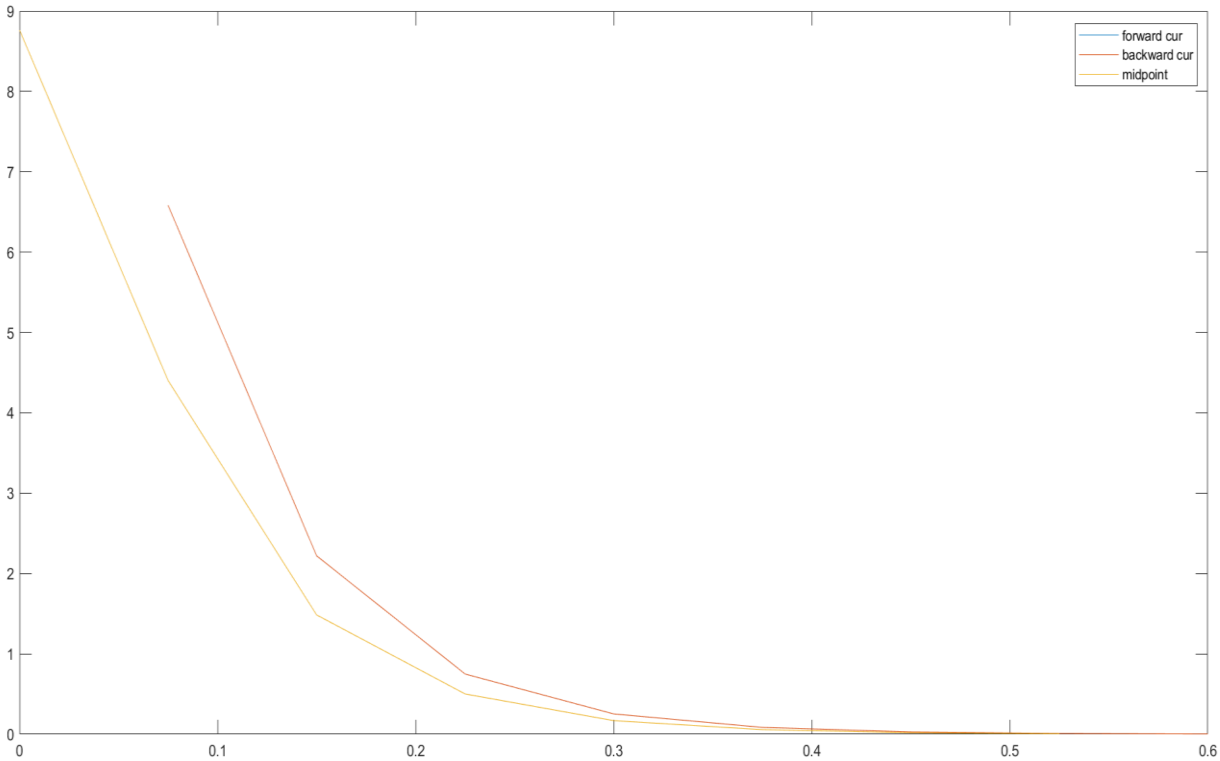
```
for i=2:1:size(current1,2)
    Voltage=(L*fdrv1(i))+(R*current1(2,i-1));
    fV_arr1(i)=Voltage;
end
fprintf('voltages of forward difference (current1.dat)\n');
volt=fV_arr1';
time=current1(1,:)';
volt_t=table(time,volt);
```

MAKING PLOT

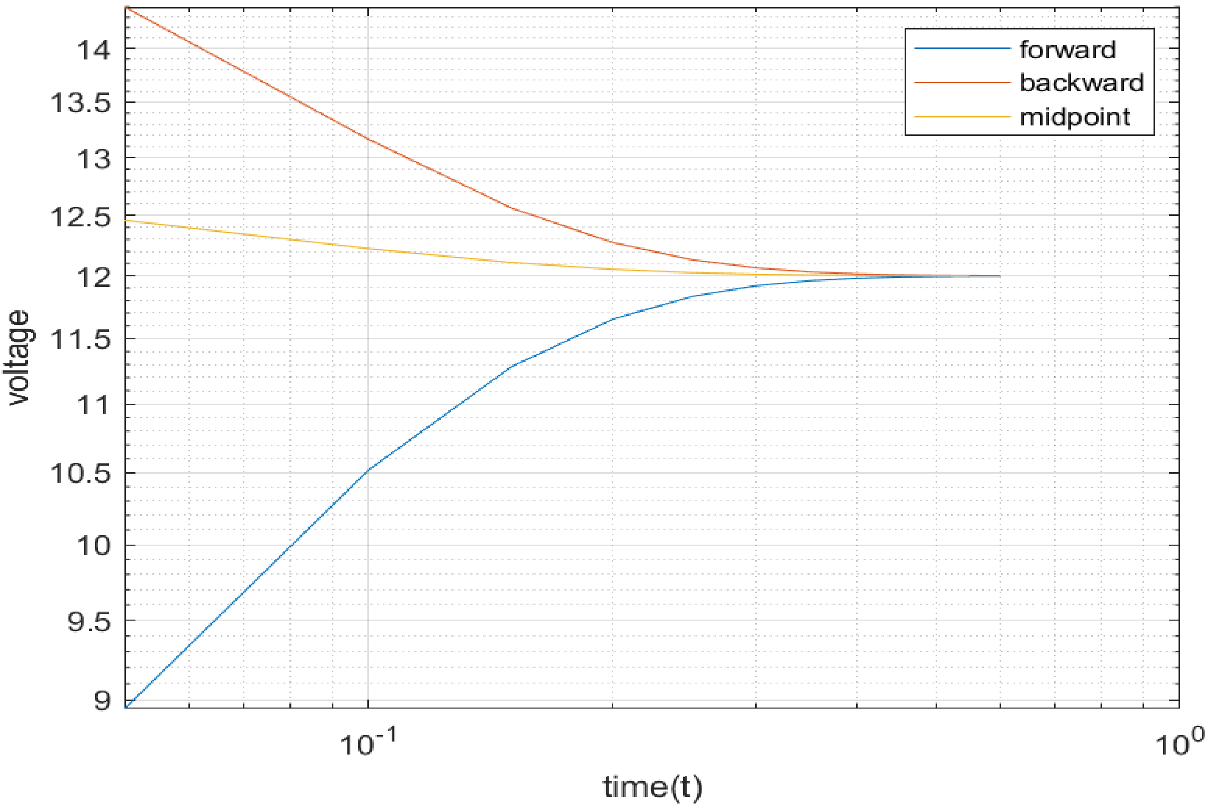
```
plot(current1(1,2:end),fdrv1(2:end),current1(1,2:end),bdrv1(2:end),current1(1,1:end-1),mdrv1(1:end-1))
legend('forward cur','backward cur','midpoint');

loglog(current1(1,2:end),fV_arr1(2:end),current1(1,2:end),bV_arr1(2:end),current1(1,1:end-1),mV_arr1(1:end-1));
legend('forward','backward','midpoint');
grid on
xlabel('time(t)');
ylabel('voltage');
```

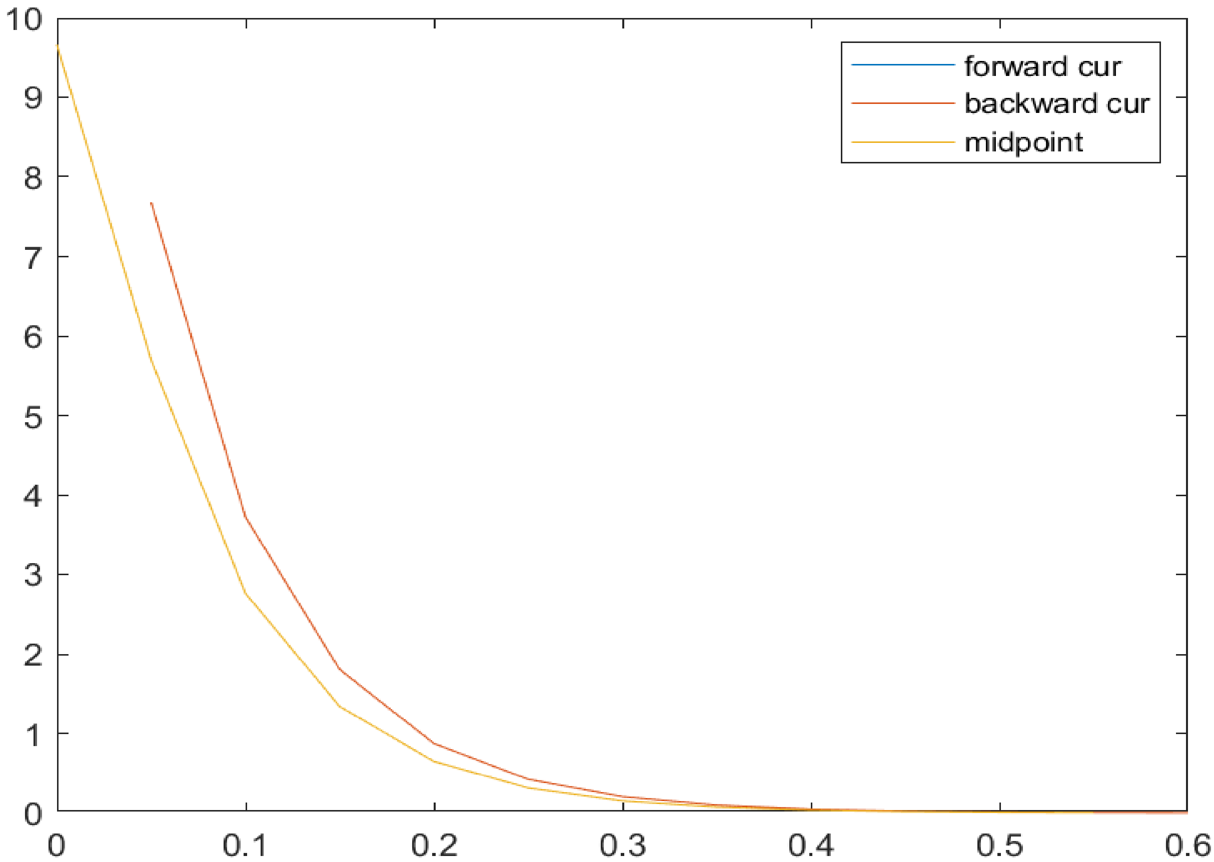
PLOTS FOR DATA-1



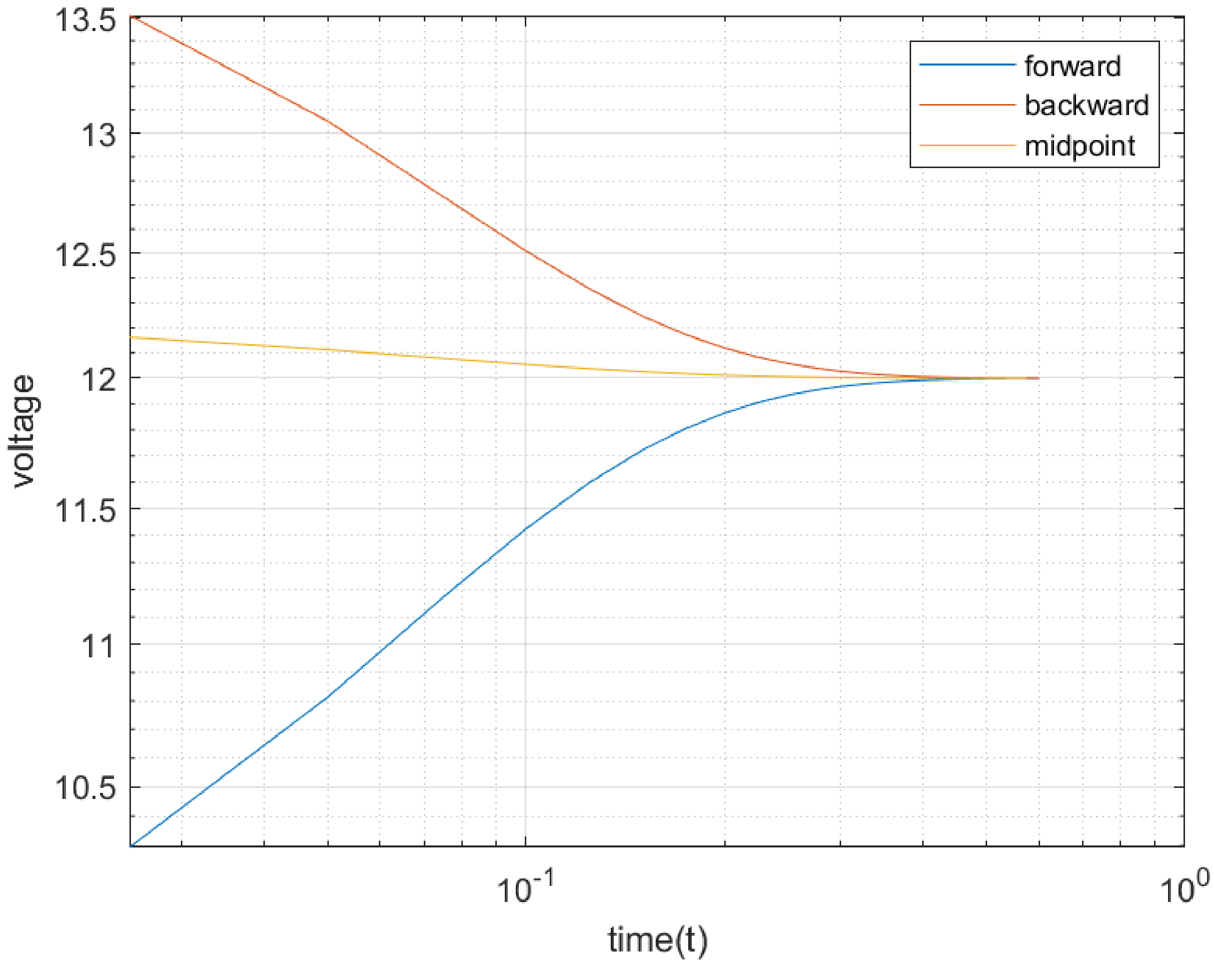


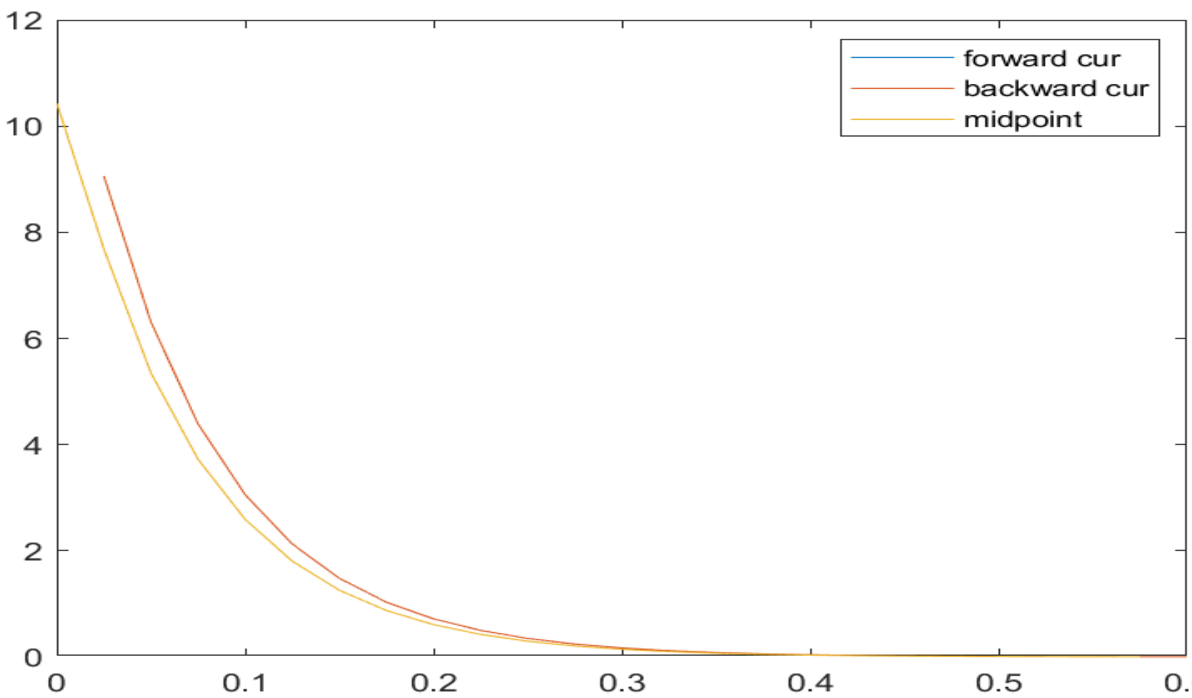
PLOTS FOR DATA-2



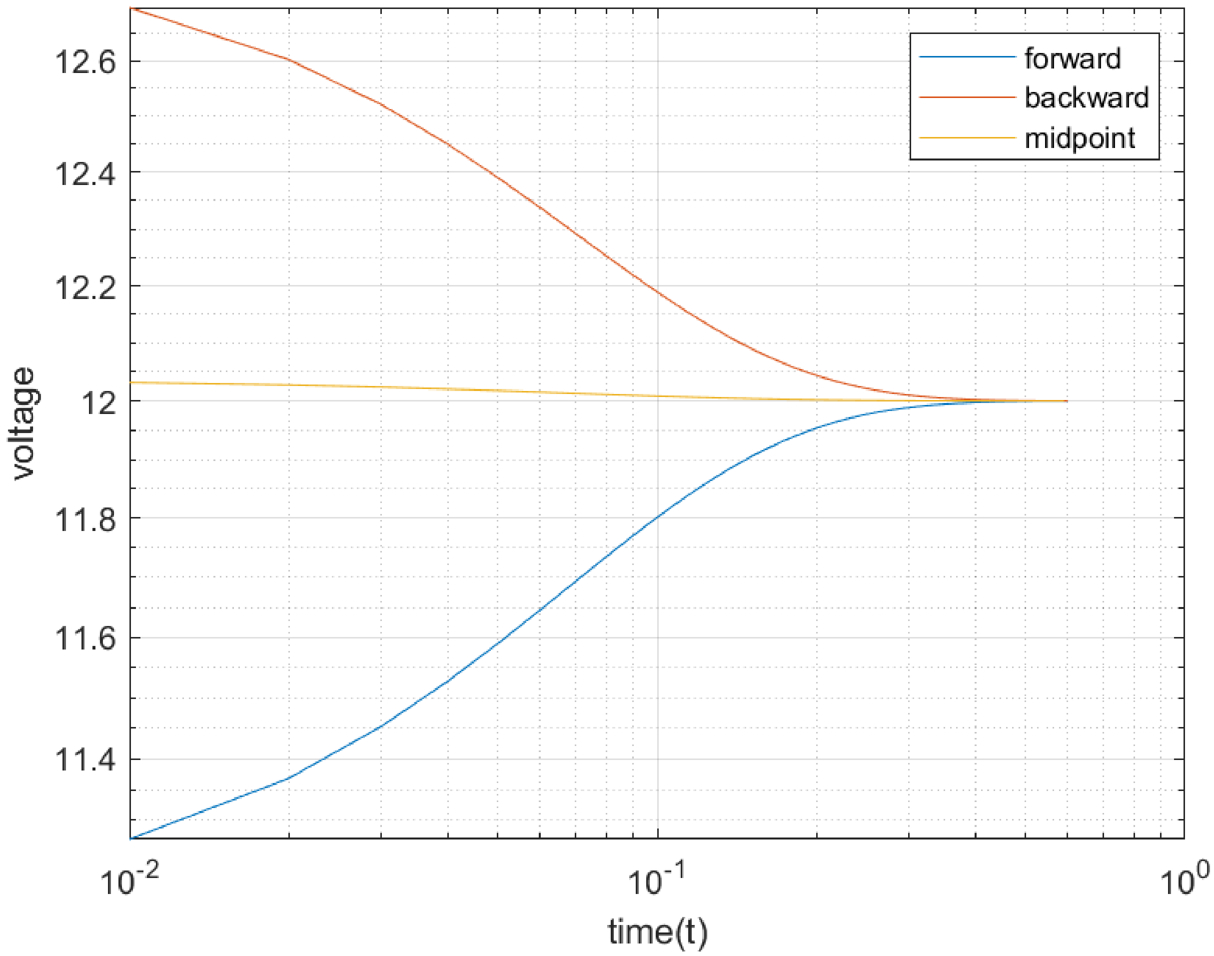


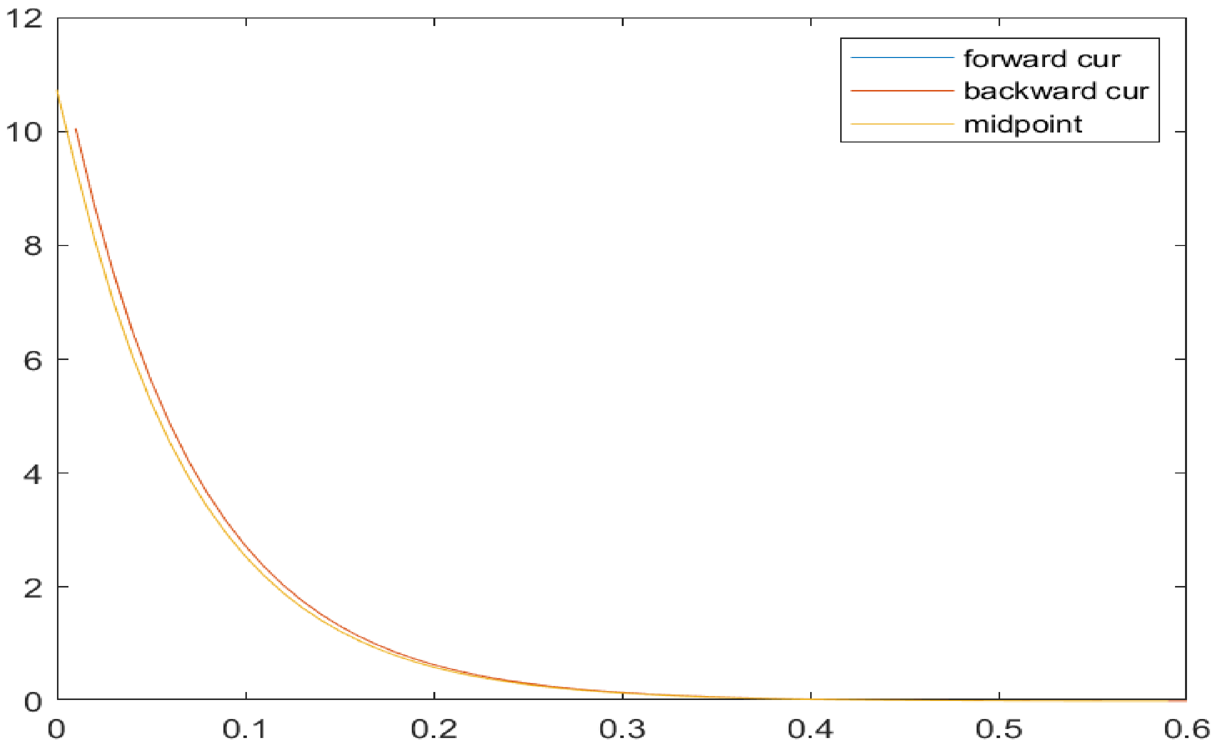
PLOTS FOR DATA-3





PLOTS FOR DATA-4





ERRORS:

```
error_forward_1 =

   0.399697825898217


error_backward_1 =

   0.348100323382591


error_midpoint_1 =

   0.059871607827767
```

```
error_forward_2 =

   0.312621907723630


error_backward_2 =

   0.286406148829291


error_midpoint_2 =

   0.048111500098605
```

```
error_forward_3 =

   0.212074548210630


error_backward_3 =

   0.203626048812763


error_midpoint_3 =

   0.079208671946388
```

```
   0.131020755693385


error_backward_4 =

   0.129024598545622


error_midpoint_4 =

   0.022128803651057
```

RESULTS:

1- As you can see on the error part the most accurate method is mid point formula second accurate method is backward difference formula the last accurate method is forward difference formula.

2- As you can see the on the error part the change of time of currents are decreasing file by file(current4.dat's h< current3.dat's h< current2.dat's h< current1.dat's h). So if you decrease the change of time of currents it gives you more accurate voltage values.

3- As you can see on the plot part the derivative of currents are approximately same each other if you compare methods for derivatives forward and backward methods have the same derivative values. Midpoint method isa little bit different from them.If you compare them by file type. Current1.dat's derivative currents plot is not so much smooth because count of Δt is less than other dat files. If you increase the count of Δt the derivative current plot is more smooth.

4- As you can see on the plot part the current1.dat's voltage plot is starting from so much far from the actual voltage but current2.dat's voltage is more closer than current1.dat. The current4.dat's initial voltage value is the closest value of the actual voltage value in every method.

Can you observe the expected convergence rates?

- Yes you can observe the converges by looking plots and voltages values.

If the endpoint formula is not used, how is the accuracy affected?

- I tried it in my code but ıt didn’t affect voltage value.

CODE:

```
clc;
clear;
format long;
file=fopen('current1.dat','r');  %files have opened..
file2=fopen('current2.dat','r');
file3=fopen('current3.dat','r');
file4=fopen('current4.dat','r');
current1=fscanf(file,'%f %f',[2 inf]);   %files have read...
current2=fscanf(file2,'%f %f',[2 inf]);
current3=fscanf(file3,'%f %f',[2 inf]);
current4=fscanf(file4,'%f %f',[2 inf]);

%current1 forward-backward-midpoint
h=current1(1,2)-current1(1,1);  % change of time of current1.dat
fdrv1=[];
for i=2:size(current1,2)             % size(current1,2)=9   so i=1 from to 9;
  der=(current1(2,i)-current1(2,i-1))/h;
  fdrv1(i)=der;
end
```

```
fV_arr1=[];
L=0.98;
R=14.2;

for i=2:1:size(current1,2)
    Voltage=(L*fdrv1(i))+(R*current1(2,i-1));
    fV_arr1(i)=Voltage;
end
fprintf('voltages of forward difference (current1.dat)\n');
volt=fV_arr1';
time=current1(1,:)';
volt_t=table(time,volt);

h=current1(1,2)-current1(1,1);
bdrv1=[];
for i=2:size(current1,2)
  der=(current1(2,i)-current1(2,i-1))/h;
  bdrv1(i)=der;
end

bV_arr1=[];
L=0.98;
R=14.2;
for i=2:size(current1,2)
    Voltage=(L*bdrv1(i))+(R*current1(2,i));
    bV_arr1(i)=Voltage;
end
```

```
fprintf('voltages of backward difference (current1.dat)\n');
volt=bV_arr1';
time=current1(1,:)';
volt_t=table(time,volt);

h=current1(1,2)-current1(1,1);
mdrv1=[];
mdrv1(1)=(-(3*current1(2,1))+(4*current1(2,2))-current1(2,3))/(2*h);
mdrv1(size(current1,2))=(current1(2,1)-4*current1(2,2)+(3*current1(2,3)))/(2*h);
for i=2:size(current1,2)-1
    der=(current1(2,i+1)-current1(2,i-1))/(2*h);
    mdrv1(i)=der;
end
mV_arr1=[];
L=0.98;
R=14.2;
mV_arr1(1)=(L*mdrv1(1))+(R*current1(2,1));
mV_arr1(size(current1,2))=(L*mdrv1(size(current1,2)))+(R*current1(2,size(current1,2)));
for i=2:1:size(current1,2)-1
    Voltage=(L*mdrv1(i))+(R*current1(2,i));
    mV_arr1(i)=Voltage;
end
fprintf('voltages of midpoint (current1.dat)\n');

volt=mV_arr1';
time=current1(1,:)';
volt_t=table(volt,time);
```

```
plot(current1(1,2:end),fdrv1(2:end),current1(1,2:end),bdrv1(2:end),current1(1,1:end-1),mdrv1(1:end-1))
legend('forward cur','backward cur','midpoint');

loglog(current1(1,2:end),fV_arr1(2:end),current1(1,2:end),bV_arr1(2:end),current1(1,1:end-1),mV_arr1(1:end-1));
legend('forward','backward','midpoint');
grid on
xlabel('time(t)');
ylabel('voltage');

% current2 forward-backward-midpoint
h=current2(1,2)-current2(1,1); %change of time of curren2.dat
fdrv2=[];

for i=2:size(current2,2)    %size(current2,2) gives you count of the currents.. so it is 13
  der=(current2(2,i)-current2(2,i-1))/h;    %derivative of current has calculated
  fdrv2(i)=der;
end
fV_arr2=[];
L=0.98;
R=14.2;
for i=2:1:size(current2,2)
    Voltage=(L*fdrv2(i))+(R*current2(2,i-1));  %voltage has calculated
    fV_arr2(i)=Voltage;
end
fprintf('voltages of forward difference (current2.dat)\n');
volt=fV_arr2';
time=current2(1,:)';
volt_t=table(time,volt);
```

```
h=current2(1,2)-current2(1,1);%0.05
bdrv2=[];

for i=2:size(current2,2)
  der=(current2(2,i)-current2(2,i-1))/h;
  bdrv2(i)=der;
end
bV_arr2=[];
L=0.98;
R=14.2;
for i=2:1:size(current2,2)
    Voltage=(L*bdrv2(i))+(R*current2(2,i));
    bV_arr2(i)=Voltage;
end
fprintf('voltages of backward difference (current2.dat)\n');
volt=bV_arr2';
time=current2(1,:)';
volt_t=table(time,volt);
h=0.050;
mdrv2=[];
mdrv2(1)=(-(3*current2(2,1))+(4*current2(2,2))-current2(2,3))/(2*h);
mdrv2(size(current2,2))=(current2(2,1)-4*current2(2,2)+3*current2(2,3))/(2*h)
for i=2:size(current2,2)-1
    der=(current2(2,i+1)-current2(2,i-1))/(2*h);
    mdrv2(i)=der;
end
```

```
mV_arr2=[];
L=0.98;
R=14.2;
mV_arr2(1)=(L*mdrv2(1))+(R*current2(2,1));
mV_arr2(size(current2,2))=(L*mdrv2(size(current2,2)))+(R*current2(2,size(current2,2)));
for i=2:size(current2,2)-1
    Voltage=(L*mdrv2(i))+(R*current2(2,i));
    mV_arr2(i)=Voltage;
end
fprintf('voltages of midpoint difference (current2.dat)\n');
volt=mV_arr2';
time=current2(1,:)';
volt_t=table(time,volt);

% plot(current2(1,2:end),fdrv2(2:end),current2(1,2:end),bdrv2(2:end),current2(1,1:end-1),mdrv2(1:end-1))
% legend('forward cur,backward cur');

% loglog(current2(1,2:end),fV_arr2(2:end),current2(1,2:end),bV_arr2(2:end),current2(1,1:end-1),mV_arr2(1:end-1))
% legend('forward','backward','midpoint');
% grid on
% xlabel('time(t)');
% ylabel('voltage');

%current3.dat
h=current3(1,2)-current3(1,1);
fdrv3=[];
```

```
for i=2:size(current3,2)
    der=(current3(2,i)-current3(2,i-1))/h;
    fdrv3(i)=der;
end

fV_arr3=[];
L=0.98;
R=14.2;
for i=2:1:size(current3,2)
        Voltage=(L*fdrv3(i))+(R*current3(2,i-1));
        fV_arr3(i)=Voltage;
end
fprintf('voltages of forward difference (current3.dat)\n');
volt=fV_arr3';
time=current3(1,:)';
volt_t=table(time,volt);

%current3.dat
h=current3(1,2)-current3(1,1); %change of time of current2.dat
bdrv3=[];

for i=2:size(current3,2) %size(current3,2) gives you count of currents..
    der=(current3(2,i)-current3(2,i-1))/h;
    bdrv3(i)=der;
end
```

```
bV_arr3=[];
L=0.98;
R=14.2;
for i=2:size(current3,2)
        Voltage=(L*bdrv3(i))+(R*current3(2,i));
        bV_arr3(i)=Voltage;
end
fprintf('voltages of backward difference (current3.dat)\n');
volt=bV_arr3';
time=current3(1,:)';
volt_t=table(time,volt);

h=0.025;
mdrv3=[];
mdrv3(1)=((4*current3(2,2))-(3*current3(2,1))-current3(2,3))/(2*h);
mdrv3(size(current3,2))=((current3(2,1))-(4*current3(2,2))+(3*current3(2,3)))/(2*h);
for i=2:size(current3,2)-1
        der=(current3(2,i+1)-current3(2,i-1))/(2*h);
        mdrv3(i)=der;
end
```

```
mV_arr3=[];
L=0.98;
R=14.2;
mV_arr3(1)=(L*mdrv3(1))+(R*current3(2,1));
mV_arr3(size(current3,2))=(L*mdrv3(size(current3,2)))+(R*current3(2,size(current3,2)));
for i=2:size(current3,2)-1
    Voltage=(L*mdrv3(i))+(R*current3(2,i));
    mV_arr3(i)=Voltage;
end

% plot(current3(1,2:end),fdrv3(2:end),current3(1,2:end),bdrv3(2:end),current3(1,1:end-1),mdrv3(1:end-1))
% legend('forward cur','backward cur','midpoint');

% loglog(current3(1,2:end),fV_arr3(2:end),current3(1,2:end),bV_arr3(2:end),current3(1,1:end-1),mV_arr3(1:end-1))
% legend('forward','backward','midpoint');
% grid on
% xlabel('time(t)');
% ylabel('voltage');

fprintf('voltages of midpoint (current3.dat)\n');
volt=mV_arr3';
time=current3(1,:)';
volt_t=table(time,volt);

%current4.dat
h=current4(1,2)-current4(1,1);
fdrv4=[];
```

```
for i=2:size(current4,2)
   der=(current4(2,i)-current4(2,i-1))/h;
   fdrv4(i)=der;
end


fV_arr4=[];
L=0.98;
R=14.2;
for i=2:1:size(current4,2)
     Voltage=(L*fdrv4(i))+(R*current4(2,i-1));
     fV_arr4(i)=Voltage;
end
fprintf('voltages of forward difference (current4.dat)\n');
volt=fV_arr4';
time=current4(1,:)';
volt_t=table(time,volt);

h=current4(1,2)-current4(1,1);
bdrv4=[];

for i=2:size(current4,2)
   der=(current4(2,i)-current4(2,i-1))/h;
   bdrv4(i)=der;
end
```

```
bV_arr4=[];
L=0.98;
R=14.2;
for i=2:1:size(current4,2)
    Voltage=(L*bdrv4(i))+(R*current4(2,i));
    bV_arr4(i)=Voltage;
end
fprintf('voltages of backward difference (current4.dat)\n');
volt=bV_arr4';
time=current4(1,:)';
volt_t=table(time,volt);
h=0.010;
mdrv4=[];
mdrv4(1)=((4*current4(2,2))-(3*current4(2,1))-current4(2,3))/(2*h);
mdrv4(size(current4,2))=((current4(2,1))-(4*current4(2,2))+(3*current4(2,3)))/(2*h);
for i=2:size(current4,2)-1
    der=(current4(2,i+1)-current4(2,i-1))/(2*h);
    mdrv4(i)=der;
end

mV_arr4=[];
L=0.98;
R=14.2;
mV_arr4(1)=(L*mdrv4(1))+(R*current4(2,1));
mV_arr4(size(current4,2))=(L*mdrv4(size(current4,2)))+(R*current4(2,size(current4,2)))
for i=2:size(current4,2)-1
    Voltage=(L*mdrv4(i))+(R*current4(2,i));
    mV_arr4(i)=Voltage;
end
```

```
fprintf('voltages of midpoint (current3.dat)\n');
volt=mV_arr4';
time=current4(1,:)';
volt_t=table(time,volt);

% plot(current4(1,2:end),fdrv4(2:end),current4(1,2:end),bdrv4(2:end),current4(1,1:end-1),mdrv4(1:end-1))
% legend('forward cur','backward cur','midpoint');

% loglog(current4(1,2:end),fV_arr4(2:end),current4(1,2:end),bV_arr4(2:end),current4(1,1:end-1),mV_arr4(1:end-1))
% legend('forward','backward','midpoint');
% grid on
% xlabel('time(t)');
% ylabel('voltage');
```

```
%errors

error_forward_1=norm(fV_arr1-12)/norm(fV_arr1);
error_backward_1=norm(bV_arr1-12)/norm(bV_arr1);
error_midpoint_1=norm(mV_arr1-12)/norm(mV_arr1);
error_forward_2=norm(fV_arr2-12)/norm(fV_arr2);
error_backward_2=norm(bV_arr2-12)/norm(bV_arr2);
error_midpoint_2=norm(mV_arr2-12)/norm(mV_arr2);
error_forward_3=norm(fV_arr3-12)/norm(fV_arr3);
error_backward_3=norm(bV_arr3-12)/norm(bV_arr3);
error_midpoint_3=norm(mV_arr3-12)/norm(mV_arr3);
error_forward_4=norm(fV_arr4-12)/norm(fV_arr4);
error_backward_4=norm(bV_arr4-12)/norm(bV_arr4);
error_midpoint_4=norm(mV_arr2-12)/norm(mV_arr4);
```